

1. Hive环境搭建

Hive: 数据仓库工具

1.1. 依赖环境

Hive不能独立运行! 需要**依赖于一个RDBMS** (帮它存储元数据) 和一个**文件系统** (帮它存储真实数据)

必须要准备一个RDBMS (也可以是hive自带的derby内嵌数据库, 不适用于生产环境, 推荐使用MySQL)
必须要准备一个Hadoop (其实也不必, 也可以使用linux文件系统来存储就行了)

1.2. 版本选择

选择任何技术的版本要考虑哪些方面: 功能, 稳定性, 可维护性, 兼容性.....

策略: **要考虑不新不旧的稳定版本**

当前 Hive 的主流版本有两个

hive-1.x, 其中使用比较多的是hive-1.2.x的版本, 最后一个稳定版本是hive-1.2.2, 如果使用这个版本, 你的hive将来没法集成spark。
hive-2.x, 现行主流的hive使用版本, 现行稳定的hive-2.x版本中的, 我们选择使用hive-2.3.6

apache-hive-2.3.6-bin.tar.gz	安装包
apache-hive-2.3.6-src.tar.gz	源码包

1.3. 自带Derby版本

用途: 测试, 学习, 快速使用

1、上传安装包

```
put -r apache-hive-2.3.6-bin.tar.gz
```

2、解压安装包

```
[bigdata@bigdata05 ~]$ tar -zxvf ~/soft/apache-hive-2.3.6-bin.tar.gz -C ~/apps/
```

3、进入到bin目录, 运行hive脚本

```
[bigdata@bigdata05 ~]$ cd apps/apache-hive-2.3.6-bin/bin
[bigdata@bigdata05 bin]$ ./hive
```

4、测试使用

```
show databases;
create database testdb;
use testdb;
create table student(id int, name string);
insert into table student values ('1','huangbo'), ('2','xuzheng'),
('3','wangbaoqiang');
select * from student;
```

1.4. 外置MySQL版本

用途：适用于企业级生产环境

第一步：准备好MySQL 和 Hadoop

请参考官网安装，或者自行安装MySQL，或者一个可用的MySQL。不管你的MySQL在哪里，只要Hive能连接上即可。如果Hive和MySQL没有安装在同一个服务器节点中，通常企业环境也都不是安装在同一个节点，一定要记得给Hive开启MySQL的远程链接权限。

要点：

- 1、有一个能用的MySQL就行
- 2、给这个MySQL配置远程连接权限

第二步：上传安装包

将 `apache-hive-2.3.6-bin.tar.gz` 到安装Hive的服务器节点中。

使用你知道的方式上传这个安装包到你的服务器上。或者使用`wget`命令直接现在去下载也可。或者给centos安装`lrzsz`命令通过自行`rz`命令来进行上传，或者通过`filezilla`这种`ftp`工具也行。

第三步：解压安装包到对应的Hive安装目录中

```
[bigdata@bigdata05 ~]$ tar -zxvf ~/soft/apache-hive-2.3.6-bin.tar.gz -C ~/apps/
```

第四步：修改配置文件

```
[bigdata@bigdata05 ~]$ cd ~/apps/apache-hive-2.3.6-bin/conf
[bigdata@bigdata05 ~]$ touch hive-site.xml
```

注意：该文件默认是没有的。一般自己创建。当然也可以从`hive-default.xml`更改而来，不推荐。

```
[bigdata@bigdata05 ~]$ vi hive-site.xml
```

在这个新创建的配置文件中加入如下截图中的内容即可：

```
<configuration>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://bigdata02:3306/myhivemetadb236?
createDatabaseIfNotExist=true&amp;verifyServerCertificate=false&amp;useSSL=false
</value>
    <description>JDBC connect string for a JDBC metastore</description>
    <!-- 如果mysql和hive在同一个服务器节点，那么请更改bigdata02为localhost -->
    <!-- bigdata02为你安装的MySQL服务器节点，请更改为自己的 -->
  </property>

  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
    <!-- SQL的驱动类名称，不用更改 -->
    <description>Driver class name for a JDBC metastore</description>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>root</value>
    <!-- 连接MySQL的用户账户名 -->
    <description>username to use against metastore database</description>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>Qwer_1234</value>
    <!-- 连接MySQL的用户的登录密码 -->
    <description>password to use against metastore database</description>
  </property>

  <property>
    <name>hive.metastore.warehouse.dir</name>
    <value>/user/hive236/warehouse236</value>
    <description>hive default warehouse, if nessecory, change
it</description>
    <!-- 可选配置，该配置信息用来指定Hive数据仓库的数据存储在HDFS上的目录 -->
  </property>
</configuration>
```

第五步：加入MySQL驱动

一定要记得加入MySQL驱动包 `mysql-connector-java-5.1.40-bin.jar` 该jar包放置在hive的安装根路径下的lib目录中，毕竟hive要读写MySQL。

```
[bigdata@bigdata05 ~]$ cp ~/soft/mysql-connector-java-5.1.40-bin.jar
~/apps/apache-hive-2.3.6-bin/lib
```

第六步：复制Hadoop集群的配置文件

一定要记得把Hadoop集群中的 core-site.xml 和 hdfs-site.xml 两个配置文件都放置在Hive安装目录下 conf目录中。

```
[bigdata@bigdata05 ~]$ cp $HADOOP_HOME/etc/hadoop/core-site.xml ~/apps/apache-hive-2.3.6-bin/conf
[bigdata@bigdata05 ~]$ cp $HADOOP_HOME/etc/hadoop/hdfs-site.xml ~/apps/apache-hive-2.3.6-bin/conf
```

第七步：安装完成，配置环境变量

```
vi ~/.bashrc
```

添加以下两行内容：

```
export HIVE_HOME=/home/bigdata/apps/apache-hive-2.3.6-bin
export PATH=$PATH:$HIVE_HOME/bin
```

最后不要忘记让环境变量配置生效：

```
[bigdata@bigdata05 ~]$ source ~/.bashrc
```

第八步：验证Hive安装

```
[bigdata@bigdata05 ~]$ hive --service version
```

第九步：初始化元数据库

注意：当使用的hive是1.x之前的版本，不做初始化也是OK的，当Hive第一次启动的时候会自动进行初始化，只不过会不会生成足够多的元数据库中的表。在使用过程中会慢慢生成。但最后进行初始化。如果使用的2.x版本的Hive，那么就必须手动初始化元数据库。使用命令：

```
[bigdata@bigdata05 ~]$ schematool -dbtype mysql -initSchema
```

```
[bigdata@bigdata05 ~]$ schematool -dbtype mysql -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/bigdata/apps/apache-hive-2.3.6-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/bigdata/apps/hadoop-2.7.7/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL: jdbc:mysql://bigdata02:3306/hivemetadb?createDatabaseIfNotExist=true&verifyServerCertificate=false&useSSL=false
Metastore connection Driver: com.mysql.jdbc.Driver
Metastore connection User: root
Starting metastore schema initialization to 2.3.0
Initialization script hive-schema-2.3.0.mysql.sql
Initialization script completed
schematool completed
[bigdata@bigdata05 ~]$
```

切记：初始化操作只需要做一次。如果你使用了一段时间，再次执行了这个命令，那么你就又得到了一个全新的Hive。

执行成功的日志：

```
[bigdata@bigdata04 hadoop]$ schematool -dbType mysql -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/bigdata/apps/apache-hive-2.3.6-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/bigdata/apps/hadoop-2.7.7/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:      jdbc:mysql://bigdata02:3306/hivemetadb2003?createDatabaseIfNotExist=true&verifyServerCertificate=false&useSSL=false
Metastore Connection Driver :   com.mysql.jdbc.Driver
Metastore connection User:     root
Starting metastore schema initialization to 2.3.0
Initialization script hive-schema-2.3.0.mysql.sql
Initialization script completed
schemaTool completed
```

第十步：启动Hive 客户端

```
[bigdata@bigdata05 ~]$ hive
[bigdata@bigdata05 ~]$ hive --service cli
```

```
[bigdata@bigdata05 ~]$ hive
which: no hbase in (/usr/local/bin:/usr/bin:/usr/sbin:/usr/sbin:/usr/local/java/jdk1.8.0_73/bin:/home/bigdata/apps/zookeeper-3.4.14/bin:/home/bigdata/apps/hadoop-2.7.7/bin:/home/bigdata/apps/hadoop-2.7.7/sbin:/home/bigdata/.local/bin:/home/bigdata/bin:/home/bigdata/apps/zookeeper-3.4.14/bin:/home/bigdata/apps/hadoop-2.7.7/bin:/home/bigdata/apps/hadoop-2.7.7/sbin:/home/hadoop/apps/apache-hive-2.3.6-bin/bin:/home/bigdata/apps/zookeeper-3.4.14/bin:/home/bigdata/apps/hadoop-2.7.7/bin:/home/bigdata/apps/hadoop-2.7.7/sbin:/home/bigdata/apps/apache-hive-2.3.6-bin/bin)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/bigdata/apps/apache-hive-2.3.6-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/bigdata/apps/hadoop-2.7.7/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/bigdata/apps/apache-hive-2.3.6-bin/lib/hive-common-2.3.6.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive>
```

第十一步：退出Hive

```
hive> quit;
hive> exit;
```

第十二步：注意事项

在使用 Hive 的时候一定要确保三件事：

- 1、确保 RDBMS元数据 启动好了。
- 2、确保 HDFS 启动好了。能正常运行
- 3、确保 YARN 集群启动好了，能正常运行

1.5. HiveServer2服务

部署方法如下：

第一：修改hadoop集群的hdfs-site.xml配置文件：加入一条配置信息，表示启用webhdfs

```
<property>
  <name>dfs.webhdfs.enabled</name>
  <value>true</value>
</property>
```

第二：修改hadoop集群的core-site.xml配置文件：加入两条配置信息：表示设置hadoop集群的代理用户

```
<property>
  <name>hadoop.proxyuser.bigdata.hosts</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.bigdata.groups</name>
  <value>*</value>
</property>
```

修改的Hadoop集群中的 core-site.xml 和 hdfs-site.xml，并且一定要记得，所有节点都的修改。重启Hadoop集群。

第三：启动Hiveserver2服务

```
[bigdata@bigdata05 ~]$ nohup hiveserver2 1>/home/bigdata/logs/hiveserver.log
2>/home/bigdata/logs/hiveserver.err &
```

第四步：启动beeline客户端

```
beeline -u jdbc:hive2://bigdata03:10000 -n bigdata
```

1.6. Hiveserver2 Web UI

Hive从2.0 版本开始，为 HiveServer2 提供了一个简单的 WEB UI 界面，界面中可以直观的看到当前链接的会话、历史日志、配置参数以及度量信息。

修改：\$HIVE_HOME/conf/hive-site.xml 配置文件：

```
<property>
  <name>hive.server2.webui.host</name>
  <value>bigdata03</value>
</property>

<property>
  <name>hive.server2.webui.port</name>
  <value>10002</value>
</property>
```

可以参考官网：<https://cwiki.apache.org/confluence/display/Hive/Setting+Up+HiveServer2#Setting+UpHiveServer2-WebUIforHiveServer2>

重启 Hiveserver2，访问 Web UI：<http://bigdata05:10002>

HiveServer2

HomeLocal logsMetrics DumpHive ConfigurationStack TraceLiap Daemons

HiveServer2

Active Sessions

User Name	IP Address	Operation Count	Active Time (s)	Idle Time (s)
bigdata	192.168.123.155	0	250	23

Total number of sessions: 1

Open Queries

User Name	Query	Execution Engine	State	Opened Timestamp	Opened (s)	Latency (s)	Drilldown Link
-----------	-------	------------------	-------	------------------	------------	-------------	----------------

Total number of queries: 0

Last Max 25 Closed Queries

User Name	Query	Execution Engine	State	Opened (s)	Closed Timestamp	Latency (s)	Drilldown Link
bigdata	select department, count(*) as total from student group by department	mr	FINISHED	27	Mon May 18 14:05:48 CST 2020	27	Drilldown

